Visualizing and Understanding Convolutional Networks

Zeiler et.al

wyuzyf，April 2019

1 Abstract

在计算机视觉任务中，卷积网络可以达到很好的效果。但是，**why they perform so well, or how they might be improved.** 本篇文章通过可视化卷积层来回答这俩个问题。

2 Introduction

卷积能达到很好的效果，有以下几个原因：（1）有大量标注好的训练图像；（2）强大的 GPU，使得可以训练大型模型；（3）更好的模型正则化策略，比如 Dropout。用可视化技术，可以揭示任意卷积层的响应，即可视化 feature maps，帮助我们找到潜在的问题。

3 Approach

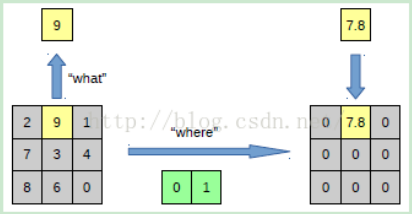
- **Visualization with a Deconvnet**

理解卷积运算需要解读中间层的特征响应，本文提出了一个将 features activity 映射回输入像素空间的方法，用到的方法是 Deconvolution Network。

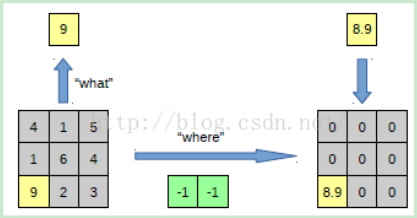
(1) Unpooling

在卷积网络中，max-pooling 是不可逆的。我们可以得到一个近似的逆通过在一组开关变量中记录区域最大值的位置。

我们知道，池化是不可逆的过程，然而我们可以通过记录池化过程中，最大激活值得坐标位置。然后在反池化的时候，只把池化过程中最大激活值所在的位置坐标的值激活，其它的值置为0，当然这个过程只是一种近似，因为我们在池化的过程中，除了最大值所在的位置，其它的值也是不为0的。刚好最近几天看到文献：《Stacked What-Where Auto-encoders》，里面有个反卷积示意图画的比较好，所有就截下图，用这篇文献的示意图进行讲解：



以上面的图片为例，上面的图片中左边表示pooling过程，右边表示unpooling过程。假设我们pooling块的大小是3*3，采用max pooling后，我们可以得到一个输出神经元其激活值为9，pooling是一个下采样的过程，本来是3*3大小，经过pooling后，就变成了1*1大小的图片了。而upooling刚好与pooling过程相反，它是一个上采样的过程，是pooling的一个反向运算，当我们由一个神经元要扩展到3*3个神经元的时候，我们需要借助于pooling过程中，记录下最大值所在的位置坐标(0,1)，然后在unpooling过程的时候，就把(0,1)这个像素点的位置填上去，其它的神经元激活值全部为0。再来一个例子：



在max pooling的时候，我们不仅要得到最大值，同时还要记录下最大值得坐标（-1，-1），然后再unpooling的时候，就直接把(-1-1)这个点的值填上去，其它的激活值全部为0。

图 1: 反池化过程

（2）Rectification

在 conv net 中，用到了 relu 非线性激活函数，这里还是用 relu。

（3）Filtering

用到了 filter 的转置，即 vertically and horizontally 方向的翻转。

conv 的过程是：input->conv filter->Activation function->feature maps

deconv 的过程是：feature maps->unpooling->Activation function->transposed filter->output

deconv 显示原图的哪一部分是不确定的，因为训练是 discriminatively。

如下图所示：

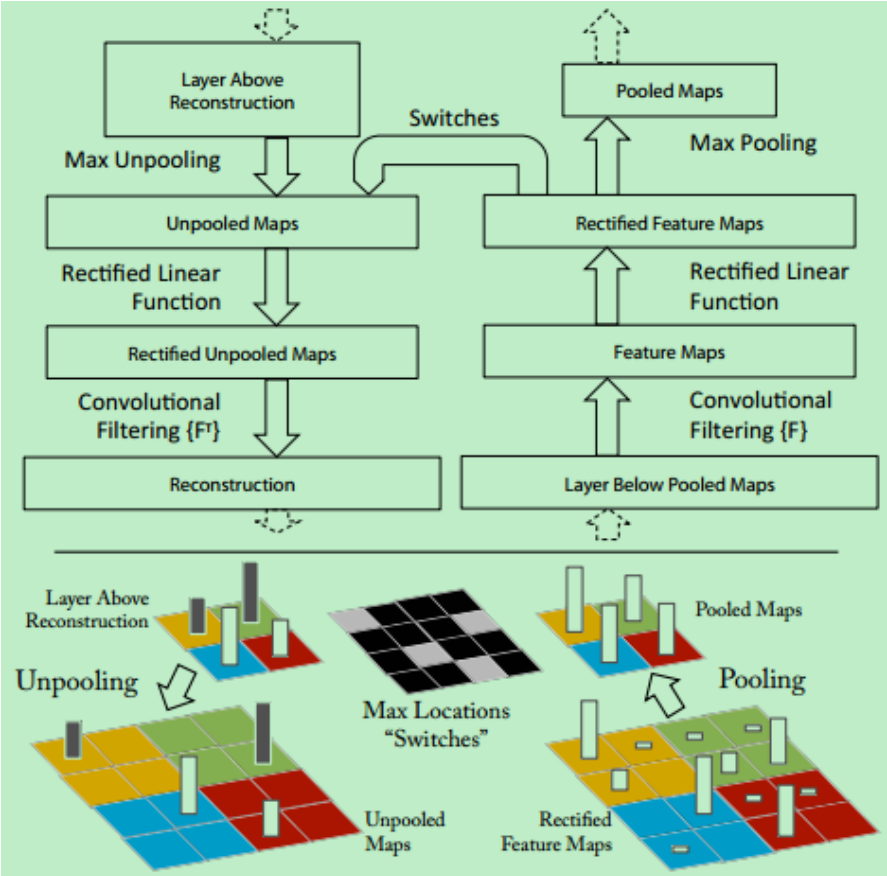


Figure 1. Top: A deconvnet layer (left) attached to a convnet layer (right). The deconvnet will reconstruct an approximate version of the convnet features from the layer beneath. Bottom: An illustration of the unpooling operation in the deconvnet, using *switches* which record the location of the local max in each pooling region (colored zones) during pooling in the convnet.

图 2: 两种卷积过程

4 Training Details

训练过程和 Alexnet 基本上相同，除了没用两个 GPU。

5 Convnet Visualization

- **Feature Visualization**

The projections from each layer show the hierarchical nature of the features in the network. Layer 2 responds to corners and other edge/color conjunctions. Layer 3 has more complex invariances, capturing similar textures (e.g. mesh patterns (Row 1, Col 1); text (R2,C4)). Layer 4 shows significant variation, but is more class-specific: dog faces (R1,C1); bird's legs (R4,C2). Layer 5 shows entire objects with significant pose variation, e.g. keyboards (R1,C11) and dogs (R4).

图 3: 特征可视化结果

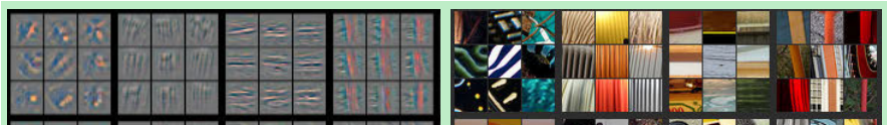


图 4: layer2

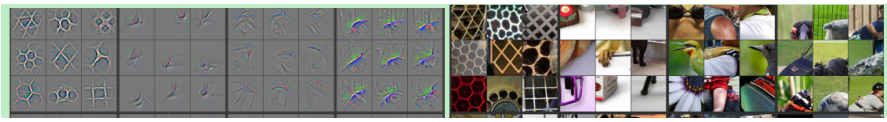


图 5: layer3

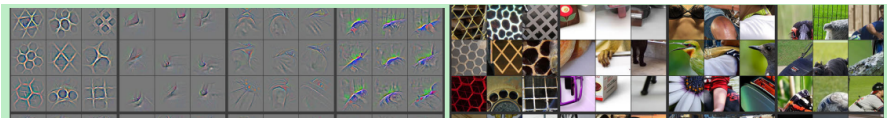


图 6: layer4

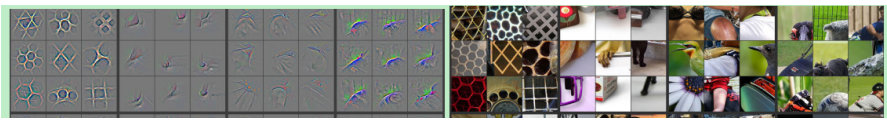


图 7: layer5

总的来说，通过 CNN 学习后，我们学习到的特征，是具有辨别性的特征，比如要我们区分人脸和狗头，那么通过 CNN 学习后，背景部位的激活度基本很少，我们通过可视化就可以看到我们提取到的特征忽视了背景，而是把关键的信息给提取出来了。从 layer 1、layer 2 学习到的特征基本上是颜色、边缘等低层特征；layer 3 则开始稍微变得复杂，学习到的是纹理特征，比如上面的一些网格纹理；layer 4 学习到的则是比较有区别性的特征，比如狗头；layer 5 学习到的则是完整的，具有辨别性关键特征。

- **Feature Evolution during Training**

需要训练到收敛的时候，高级特征才能有好的表现。

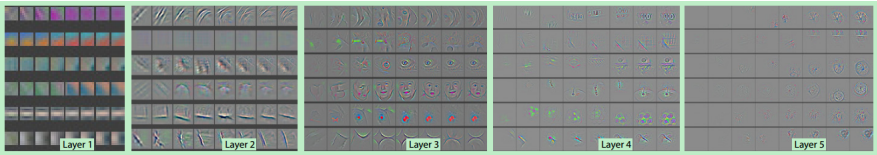


Figure 4. Evolution of a randomly chosen subset of model features through training. Each layer's features are displayed in a different block. Within each block, we show a randomly chosen subset of features at epochs [1,2,5,10,20,30,40,64]. The visualization shows the strongest activation (across all training examples) for a given feature map, projected down to pixel space using our deconvnet approach. Color contrast is artificially enhanced and the figure is best viewed in electronic form.

图 8: 特征可视化

作者给我们显示了，在网络训练过程中，每一层学习到的特征是怎么变化的，上面每一整张图片是网络的某一层特征图，然后每一行有 8 个小图片，分别表示网络 epochs 次数为：1、2、5、10、20、30、40、64 的特征图：

结果：(1) 仔细看每一层，在迭代的过程中的变化，出现了 sudden jumps;(2) 从层与层之间做比较，我们可以看到，低层在训练的过程中基本没啥变化，比较容易收敛，高层的特征学习则变化很大。这解释了低层网络的从训练开始，基本上没有太大的变化，因为梯度弥散嘛。(3) 从高层网络 conv5 的变化过程，我们可以看到，刚开始几次的迭代，基本变化不是很大，但是到了 40 50 的迭代的时候，变化很大，因此我们以后在训练网络的时候，不要着急看结果，看结果需要保证网络收敛。

- **Feature Invariance**

sample images being translated, rotated and scaled by varying degrees, 小的变换在第一层有比较大的影响，但在后面几层有较小的影响。通常情况下，cnn 是没有旋转和尺度不变形的。

个人总结：我个人感觉学习这篇文献的算法，不在于可视化，而在于学习反卷积网络，如果懂得了反卷积网络，那么在以后的文献中，你会经常遇到这个算法。大部分 CNN 结构中，如果网络的输出是一整张图片的话，那么就需要使用到反卷积网络，比如图片语义分割、图片去模糊、可视化、图片无监督学习、图片深度估计，像这种网络的输出是一整张图片的任务，很多都有相关的文献，而且都是利用了反卷积网络，取得了牛逼哄哄的结果。所以我觉得我学习这篇文献，更大的意义在于学习反卷积网络。